Network Centric Architectures: Are We Up To The Task?

Stuart T. Booth
Vitech Corporation
2070 Chain Bridge Road, Suite 100
Vienna, VA, USA 22182
sbooth@vitechcorp.com

Abstract. *The emergence of network centric architectures and the achievement of information superiority is the new paradigm that is being embraced by the military's next generation of systems to be developed and deployed. The changes dictated in this new architecture are instigating revolutionary changes throughout the DoD. These changes should have an equally profound effect on system integrators in terms of integration practices and policies that will allow network centric architectures to be realized within budgetary and schedule time constraints. The purpose of this paper is to identify key issues currently limiting the effectiveness of system integrators in their efforts to architect network centric architectures and offer suggestions on how to strengthen the integration process through the application of model-based systems engineering, integrated information repositories, and teams that have both a vertical and horizontal architecture definition and integration responsibilities.*

Background

The emergence of network centric architectures and the achievement of information superiority is the new paradigm being embraced by the military's next generation of systems to be developed and deployed in support of the Department of Defense (DoD) Joint Vision 2010. The changes dictated by this new architecture are instigating revolutionary changes throughout the DoD. These changes should have an equally profound effect on system integrators in terms of integration practices and policies that will allow network centric architectures to be realized within budgetary and schedule constraints.

A network centric architecture is a revolutionary strategy intended to transform the DoD's organization, processes, doctrine, tactics, and system acquisition decisions to realize dominance on the battlefield. The far-reaching results of such a change might be compared to what the Internet and e-business have done in the commercial marketplace. Traditional stove-pipe organizations, systems, and processes will be adapted (or replaced) with horizontal processes and agile organizations that will require a level of integration never before witnessed. The realization of such an architectural approach will be a supreme challenge for the DoD, their coalition partners, and industry. Currently, two notable network centric programs are underway with the U.S. Army's Future Combat Systems (FCS) and the Navy's FORCEnet. These programs are expected to be initially fielded in the next several years, with incremental upgrades to be employed well into the future.

The following section provides a brief overview of a network centric architecture and why such an architecture is considered a complex system that requires the application of our industry's best practices to effectively develop, integrate, and field these new architectures.

A network centric architecture is enabled by three primary elements:

1. **Information Grid:** Forms the basis for information superiority that consists of network of networks, which are made up of communication nodes, operating systems, and information management applications that enable network centric computing across the Joint Battle space. The connectivity and computing capabilities of the information grid enable the sensor grid to generate battlespace awareness, a key building block of information superiority.
2. **Sensor Grid:** Consists of an array of sensors that cover the battlespace including air, sea, ground, space, cyberspace, and dedicated platform sensors. These sensors are managed and directed to provide battlespace information, including platform and soldier information, which allows the Joint force to increase battlespace awareness and synchronize battlespace awareness with military operations.

3. **Engagement Grid:** Provides effective decision making and coordination support that exploits battlespace awareness enabled through predictive planning and pre-emption, integrated force management, and execution of time-critical missions.

These network centric architecture elements are used to:

- Effectively utilize combat power to be responsive, accurate, and lethal
- Execute operations decisively and with a high operational tempo
- Shape the battlespace
- Lock out enemy courses of action.

A network centric architecture is composed of physical elements commonly referred to as a System of Systems (SoS) meaning that individual systems (i.e., planes, tanks, and ships) and the people, processes, and facilities are linked together via a network that comprises the entire architecture. An example network centric architecture is represented in Figure 1, where the arrows on the diagram represent possible communications exchanges between the respective system elements. The implication of this interaction multiplied hundreds or thousands of times results in a complex web of interfaces between the system elements that participate within the network to achieve a common goal. It is beyond the scope of this paper to explain how this example architecture is intended to work; however, the point is that for a given network centric architecture, the operational organization, systems, and processes must be explicitly synchronized to achieve the desired results, and that everything about this architecture must be methodically engineered from end to end.

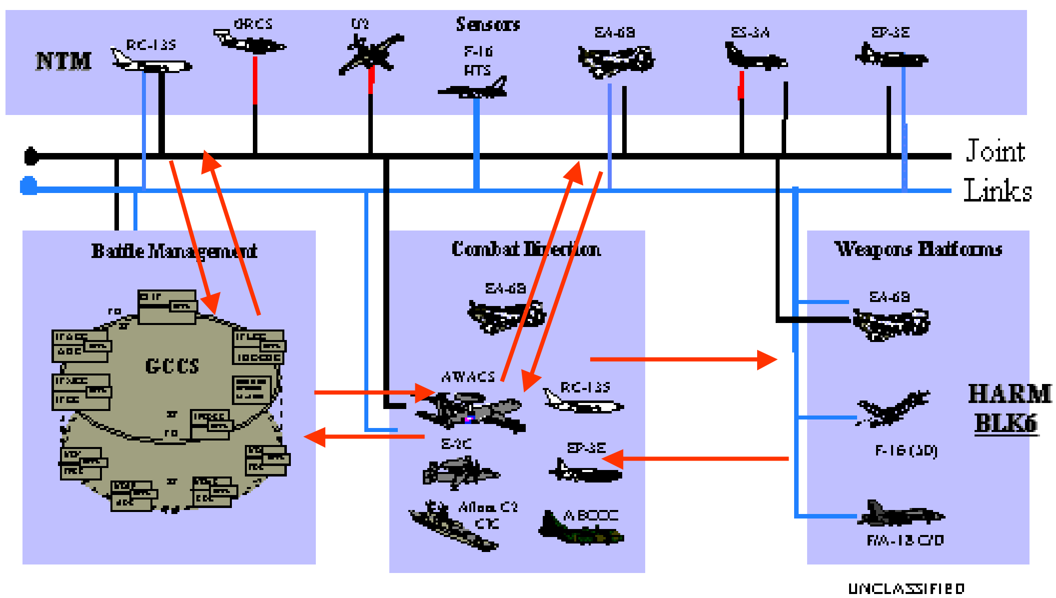

Figure 1. Example Network Centric Architecture (Source: JCS undated)

Product and Process Integration

The ability to produce quality products based on quality processes is well understood and actively supported by such organizations as the Software Engineering Institute (SEI), International Organization for Standardization (ISO), and other industry groups that promote process improvement. System integrators take great pride in achieving high levels of process maturity because it represents the capability of producing high quality products and positions their company as being a serious competitor in the marketplace. While achieving high levels of process maturity certainly indicates that an organization is well positioned to perform effectively, current standards and capability maturity models provide vague reference to key disciplines and practices that are needed to effectively define, develop, and integrate network centric architectures.

The challenges are compounded by the nature of large defense programs today, which requires bridging organizations, processes, and competencies into a homogeneous integration entity. Recognizing this challenge, the government has established the concept of the Lead Systems Integrator (LSI) responsible for integrating the desired system. The LSI is supported by an array of specialty suppliers that are typically large system integrators themselves. This yields a complex web of relationships between companies that have different cultures, methods, and processes for system integration. The LSI has the responsibility of organizing these relationships and establishing complementary and unified methods, processes, and practices that are best suited for the problem at hand. Ultimately, if there are problems with aligning organizations with their processes, methods, and tools within the engineering process, then there will be communication, process, and product disconnects that increase the risk of successful integration. The consequences of this risk results in the likely probability of increased cost and schedule to the program. While the quantification of this risk is difficult to assess, the implication of it is evident based on the nature of network centric architectures and the high degree of interaction between the systems. For the class of systems integrated within a network centric architecture, current integration practices require strengthening to effectively integrate such architectures. A key integration practice that requires strengthening is the ability to integrate the requirements and architecture efforts into a seamless model. A solution to this problem is to apply model-based systems engineering practices.

For the purpose of this paper, the definition of model-based systems engineering is the application of scientific and engineering efforts to transform an operational need into a description of system performance parameters and a system configuration by creating executable, explicit representations (model) of a system in order to predict, simulate, and explain the resultant behavior of the system from the structure.

As depicted in Figure 2, model-based systems engineering consisting of requirements analysis, behavioral analysis, architecture synthesis, verification, and simulation are performed in an integrated manner (where the lines signify traceability). Equal emphasis is placed in each of these areas ensuring that the design is complete and consistent within each layer of the architecture, starting from top (layer 1) down to the lowest layer (layer N) in the architecture. As the architecture evolves, the benefits of model-based systems engineering become fully realized by having the means to truly assess impacts not only from a static traceability perspective, but also from a dynamic perspective in the form of function execution and resource utilization.

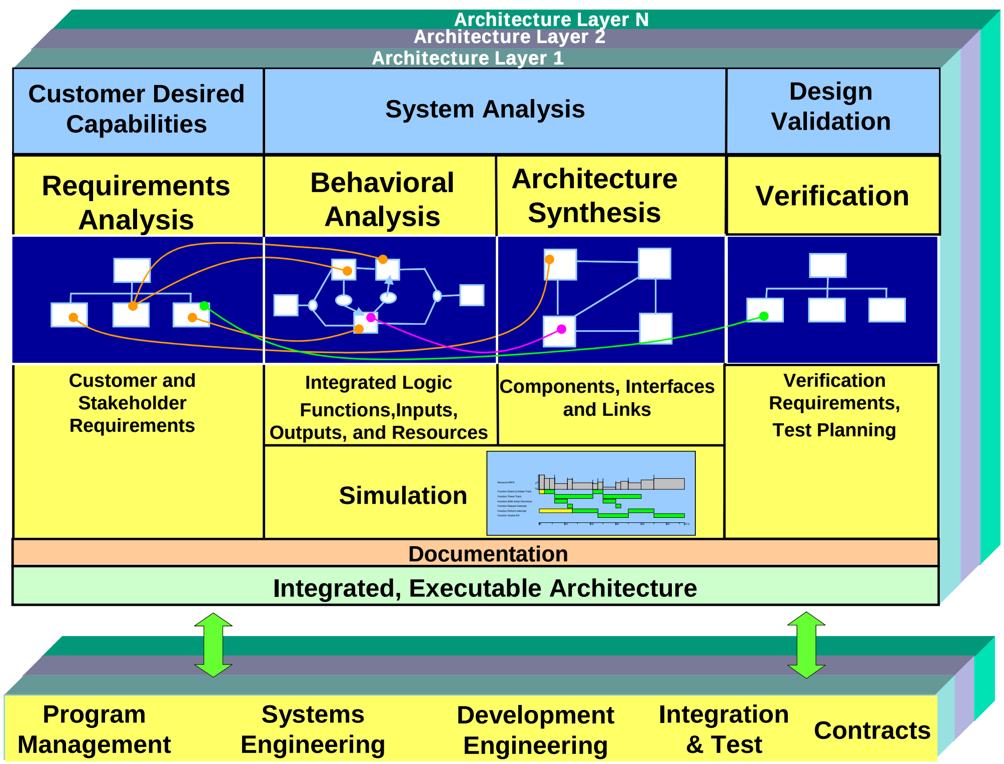

Figure 2. Model-Based Systems Engineering

To take the concept of model-based system engineering further, EIA 632 (ANSI/EIA-632-1998) is used as a basis for describing what an integrator does in developing a network centric architecture. This is a comprehensive standard that defines an explicit set of requirements to be satisfied by a system developer. As a notional model, the "V" diagram is shown in Figure 3; the top-down and bottom-up process represents the realization of a system within the development lifecycle. The execution of this process represents the refinement of the solution from layer to layer until the solution products are completely defined and developed. At the conclusion of the process, the customer ultimately receives a tested and validated system. The figure has been augmented to include model-based systems engineering practices as being an essential part of the process in developing network centric architectures. Each model defines the network (end-to-end) system threads, which establish the definition of communications between the elements at a particular system layer. As the system is decomposed from layer 1 to layer 2 and so on, the system threads are also decomposed, while preserving the inputs, outputs, and performance requirements from the previous layer.

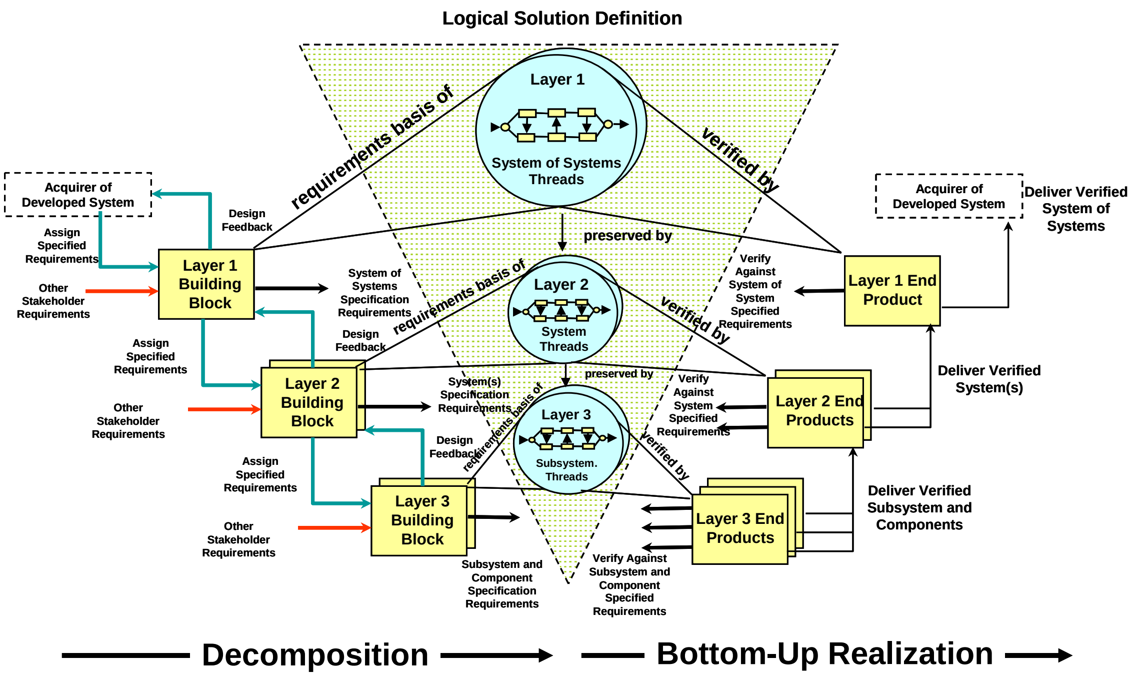

Figure 3. SoS Development using MBSE

As a general practice, defining system threads is pretty well understood. While aggressive application of UML (OMG 2003) is being applied on a wide scale, bridging the UML models from one architecture layer to the next has proved difficult, thus providing opportunities for missing interfaces and functions that don't get propagated through the development process. Another challenge the UML approach has demonstrated is the difficulty in associating the UML use-case activity models to requirements. Current UML approaches don't support requirements traceability below the use-case level, which adds additional problems in developing architecture based specifications. The combined consequences of these two issues results in human intensive efforts to maintain consistency of the UML architecture definition between the architecture layers and the requirements.

What is needed is an integrated schema that supports architecture decomposition and requirements traceability. Model-based systems engineering practices address this problem as shown in Figure 4, using Vitech Corporation's CORE model-based systems engineering software tool. Figure 4 provides an information model that demonstrates traceability between requirements, behavior (*Functions* and *Items*) and physical architecture views (*Components*, *Links* and *Interfaces*), thus insuring consistency between the requirements and the architecture, both within a layer and between layers. As a result, this approach overcomes a key risk factor of disassociation between requirements and architecture currently experienced in applying UML on network centric architecture projects.

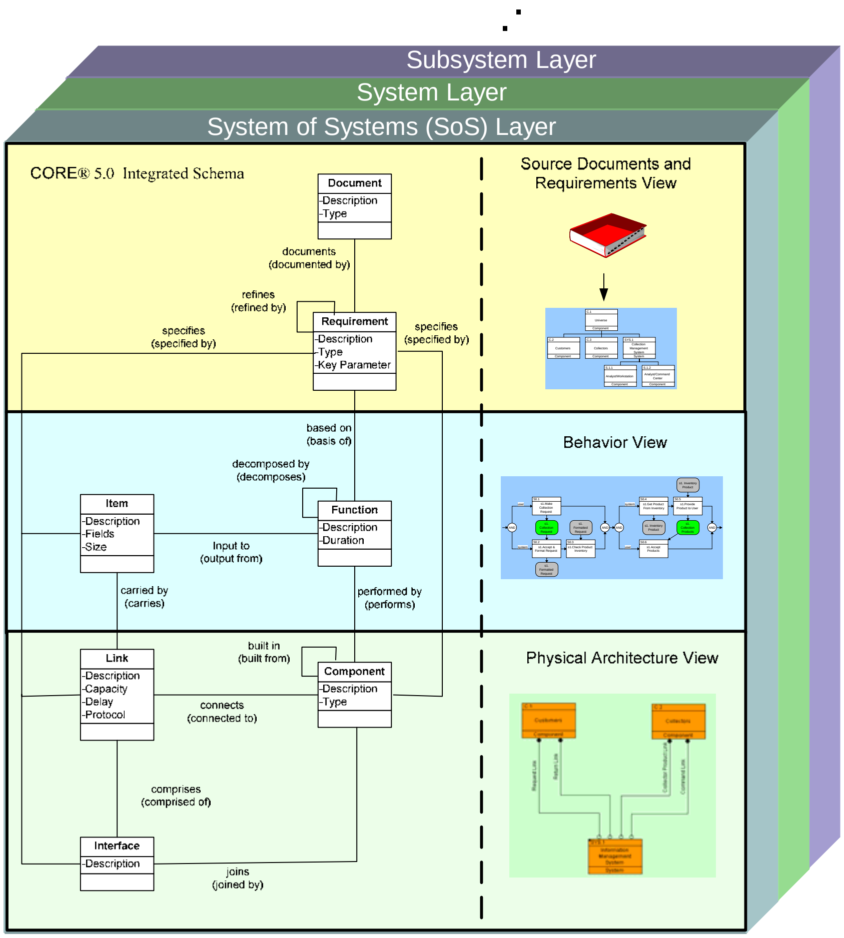

Figure 4. CORE Integrated Schema

In addition to the importance of traceability between requirements and architecture models are the properties of the models themselves. Figure 5 provides a simplified example of an executable system thread utilizing the Enhanced Functional Flow Block Diagram (EFFBD) notation. The model semantics consist of concurrency (AND symbol with branches) with *Functions* and *Items* that form the system thread. The model is performed from left to right with each *Function* generating *Items* that represents network information exchanges. Each *Function* and *Item* contains its own attributes and relations (as shown in Figure 4) that form the specification of the system thread. This specification requires traceability to requirements and allocation of the *Functions* to *Components* representing the network physical elements. This analysis approach is performed at each layer in the architecture, with each level of decomposition preserving the previous layer. To further illustrate the model-based systems engineering approach, the system thread also models performance by identifying the loop response requirement from the initial trigger to the expected response time. When the model is executed, a function timeline is generated validating the loop response time. As the architecture is decomposed, engineering trade-off decisions are made regarding functional partitioning and allocation of the behavior. Model-based systems engineering takes this into account through the preservation of performance decomposition to ensure loop response times are preserved at each layer of the architecture.

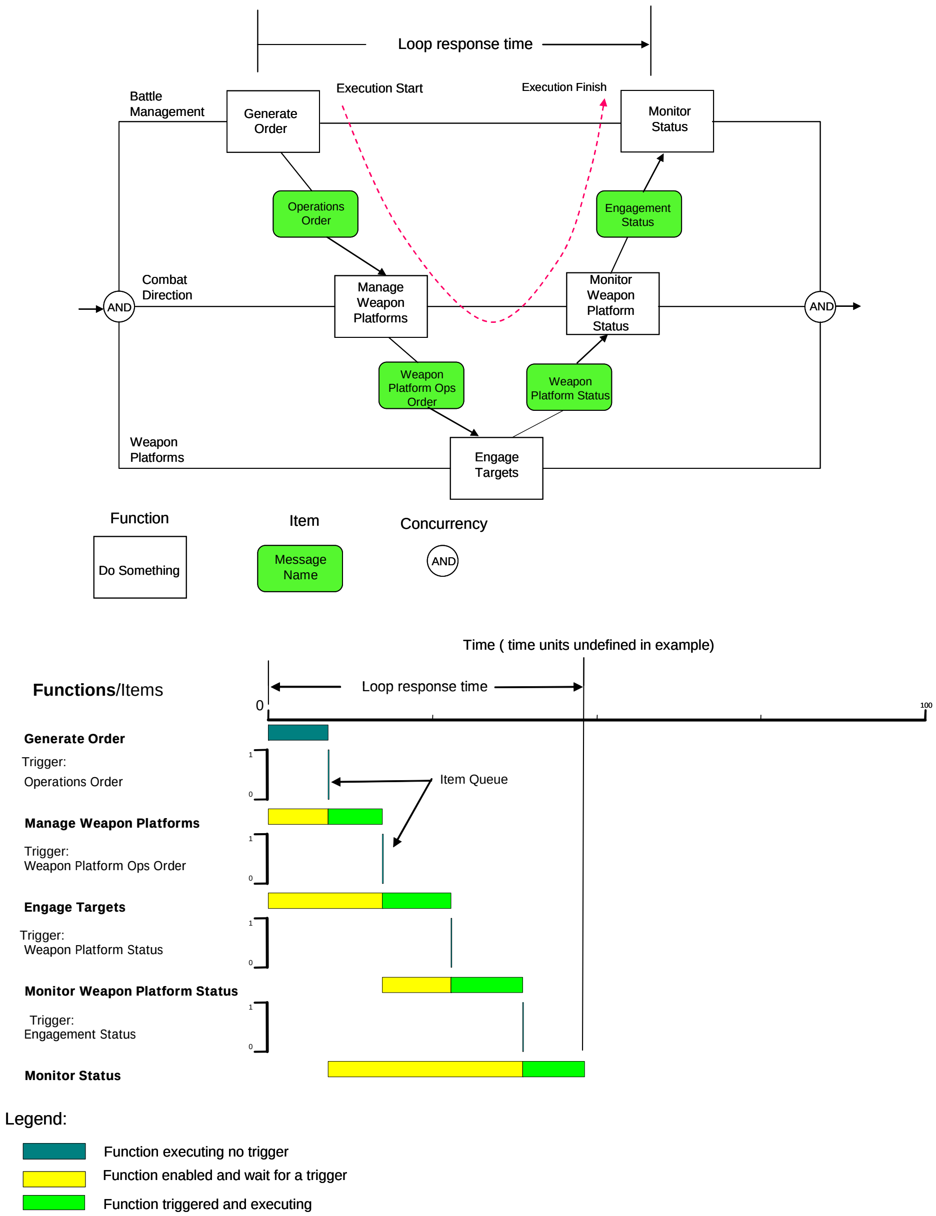

Figure 5. EFFBD Executable Behavior Model Example

Vertical Engineering Practices and Implications

Figure 6 illustrates another challenge we face in developing and integrating network centric architectures. Historical system development practices, organizational inertia, narrowly defined engineering competencies, and specialized tools have contributed to the establishment of institutionalized structures that create organizational and information stove pipes. The integration of complex systems with the degree of horizontal processes that link together systems within a network centric architecture makes it very difficult to maintain the integrity of the design as it is decomposed from layer to layer. In this illustration, a typical System of Systems architecture consists of system of systems, systems, subsystems, and components (components not shown in diagram for simplification). At each layer, organizations such as requirements engineering, architecture teams, verification teams, and a number of specialty engineering and design teams are involved in the development of the system. Each of these teams have their own repositories, some of which may be posted on a general data repository, but access and visibility to the data is still difficult. The stove-piping of information and insufficient coordination leads to locally optimized solutions, resulting in incompatible interfaces and a design that can't be integrated. The path forward is to apply model-based systems engineering practices with an integrated data repository.

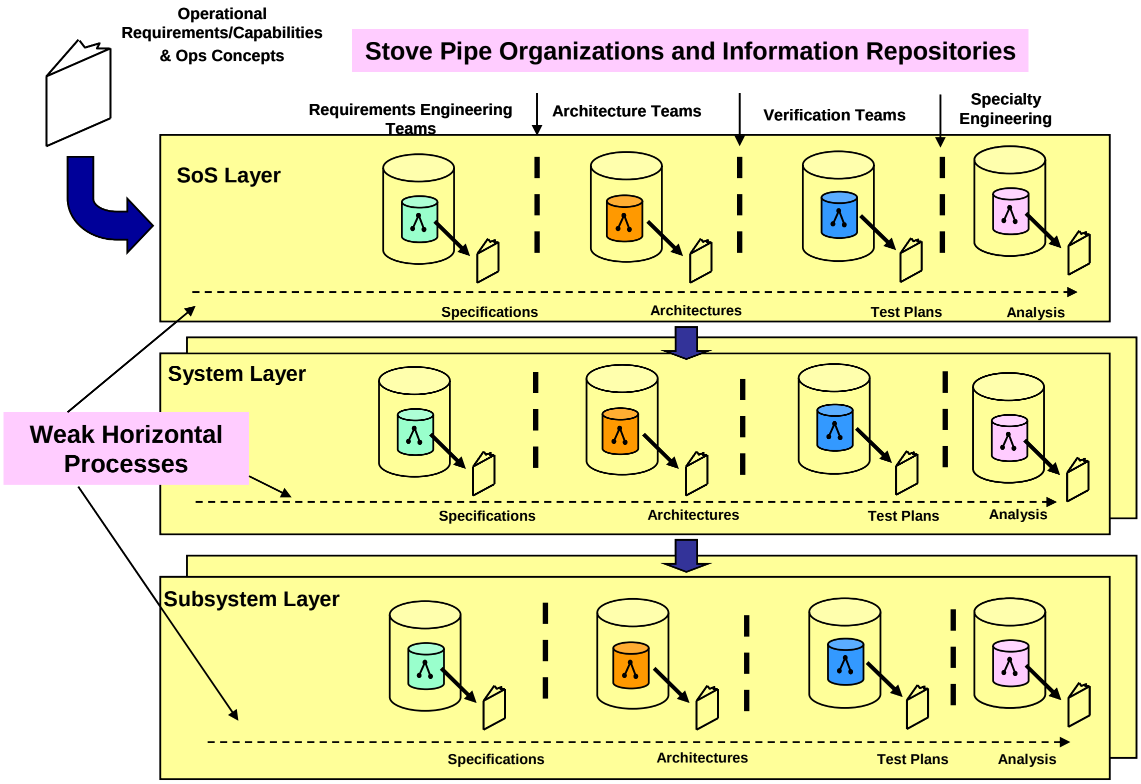

Figure 6. Current Engineering and Repository Stove Pipes

Figure 7 is a depiction of a possible solution for effectively developing network centric architectures. This solution is based on the idea of having both horizontal and vertical integration teams applying model-based systems engineering practices within a seamless model-based information environment. The horizontal teams (Functional IPTs) are responsible for defining the network services and allocating the functions and performance requirements across the vertical systems architecture. The system teams are responsible for developing and delivering the hardware and software system elements that perform the functions and achieve the desired capabilities. The engineering artifacts that are generated from the common design repository are consistent and validated through traceability and simulation at each layer of the architecture. As a result, this approach overcomes many of the challenges that teams face today with fragmented information sources that lead to inconsistencies. This approach provides a clear integration between requirements and the architecture development processes. The end result of applying model-based systems engineering within an integrated environment coupled with teams that have both vertical and horizontal responsibilities provides the essential ingredients to overcome the integration silos that hamper effective integration.

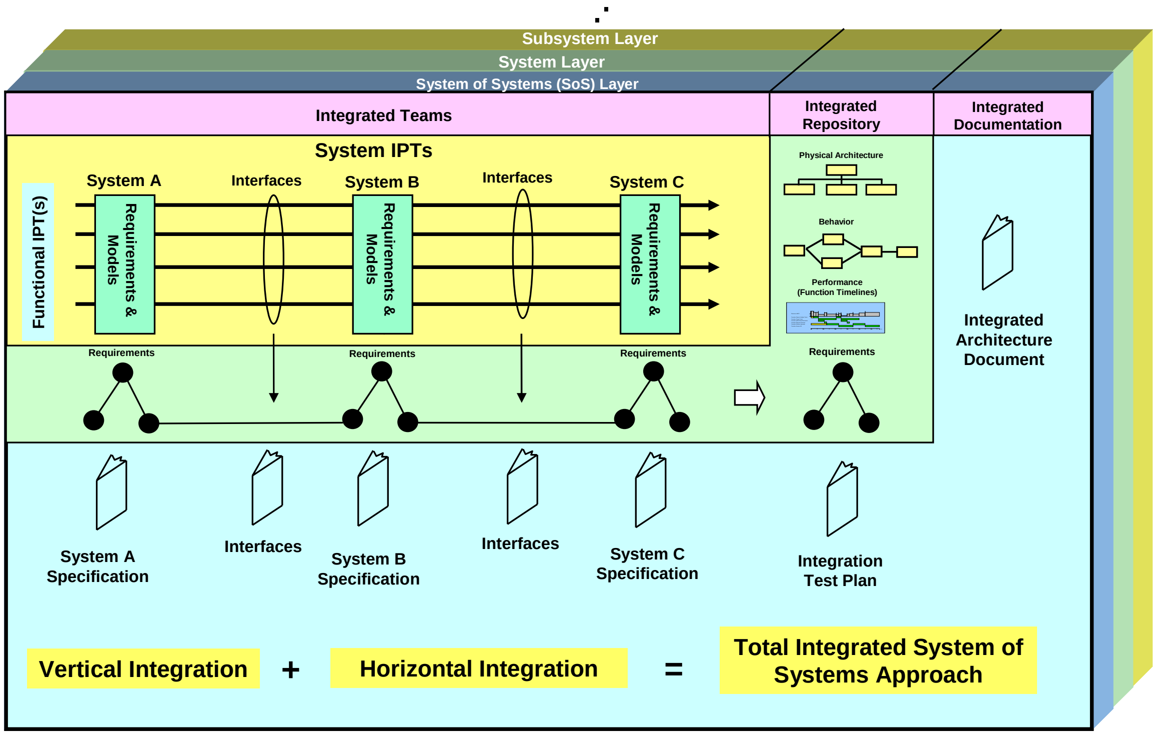

Figure 7. Total Integrated System of Systems Approach

The following is a summary of recommended actions to strengthen engineering practices that augment accepted standards in support of effective integration of network centric architectures:

- Address the horizontal and vertical integration practices equally. Horizontal integration is responsible for end-to-end functional integrity, and vertical integration is responsible for end-item integration. Collaboration between the two integration focuses is essential.
- Institute model-based integrated information repositories that will allow effective collaboration and integration between the horizontal and vertical integration efforts.
- Apply engineering methodologies that support effective modeling of the architecture both horizontally and vertically. The modeling method must support the definition of end-to-end system threads, preserve decomposition, be executable, and provide the capability to associate requirements to the elements within the behavior model.
- Train engineers in model-based systems engineering supported by appropriate tools.
- Integrate executable system behavior models as an integral part of the specification development process.
- Strengthen systems engineering standards and maturity models to address model-based systems engineering practices as described in this paper.
- Employ the use of effective collaboration technologies that allow dispersed teams to communicate and share information and knowledge.

Conclusion

Current integration practices of separating requirements from architecture development, coupled with weak horizontal integration practices, represent some of the key issues in effectively defining and integrating network centric architectures. Model-based systems engineering practices, when performed by integration teams that have both strong horizontal and vertical integration responsibilities, and with an integrated repository offer a promising vision on how the current challenges can be overcome. Instituting these practices comes at a price. Investments must be made in developing competencies in the integration workforce. These competencies include the application of model-based systems engineering practices, tools, and information repositories that enable a seamless integration environment. Ultimately, the work must be performed. Infrastructure expenses will be offset by reduced rework during integration, thereby delivering higher functionality from the customer's dollars. Lastly, engineering standards and capability maturity models must provide reinforcement, best practices, and guidance to the integrators as they invest in integration processes, tools, and information repositories.

References

ANSI/EIA-632-1998, Processes for Engineering a System.

DoD Architecture Framework Working Group, *DoD Architecture Framework Version 1.0.*

U.S. Department of Defense, Arlington, Virginia, 9 February 2004.

Joint Chiefs of Staff J6 C4, *Observations on the Emergence of Network-Centric Warfare* Information paper, http://www.dtic.mil/jcs/j6/education/warfare.html

Object Management Group (OMG), *Unified Modeling Language Specification*, March 2003, Version 1.5

US Department of Defense, *Joint Vision 2010*.

NMCI Industry Symposium, *FORCEnet – Engineering & Architecting the Navy's IT Future* Presentation, RADM Mike Sharp, USN, Vice Commander, Space & Naval Warfare Systems Command, 19 June 2003

Biography

Mr. Booth is a Principal Systems Engineer at Vitech Corporation. He has more than 16 years of experience in the defense, telecommunications, and automotive industries. He has extensive international experience in large-scale systems integration projects managing cross functional teams performing systems engineering, business process engineering and integration and test.

Mr. Booth received a B.S. degree in Electrical Engineering and Computer Science from the University of California at Berkeley. He also has the distinction of being a member of the US Navy crew who commissioned the first Trident submarine, USS Ohio SSBN 726.

This paper was presented at the International Council on Systems Engineering 15th Annual International Symposium (Rochester, NY, 10-14 July 2005) and is included in the proceedings available at www.incose.org.